

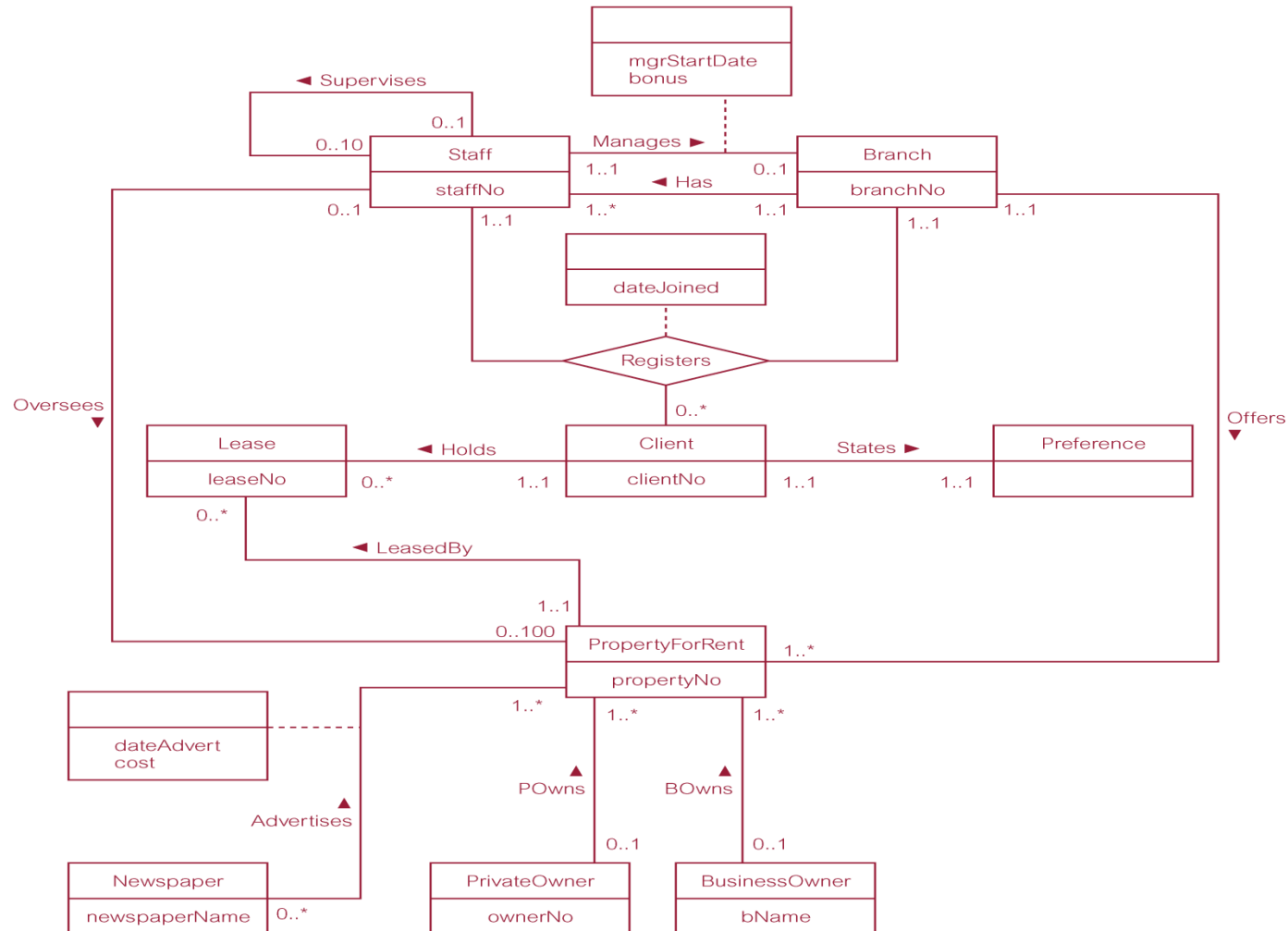
Chapter 4

ENTITY RELATIONSHIP DIAGRAM

Chapter 4 - Objectives

- ▶ **How to use Entity–Relationship (ER) modeling in database design.**
- ▶ **Basic concepts associated with ER model.**
- ▶ **How to build an ER model from a requirements specification.**

ER Diagram of Branch View of *DreamHome*



Concepts of the ER Model

- ▶ **Entity types**
- ▶ **Relationship types**
- ▶ **Attributes**

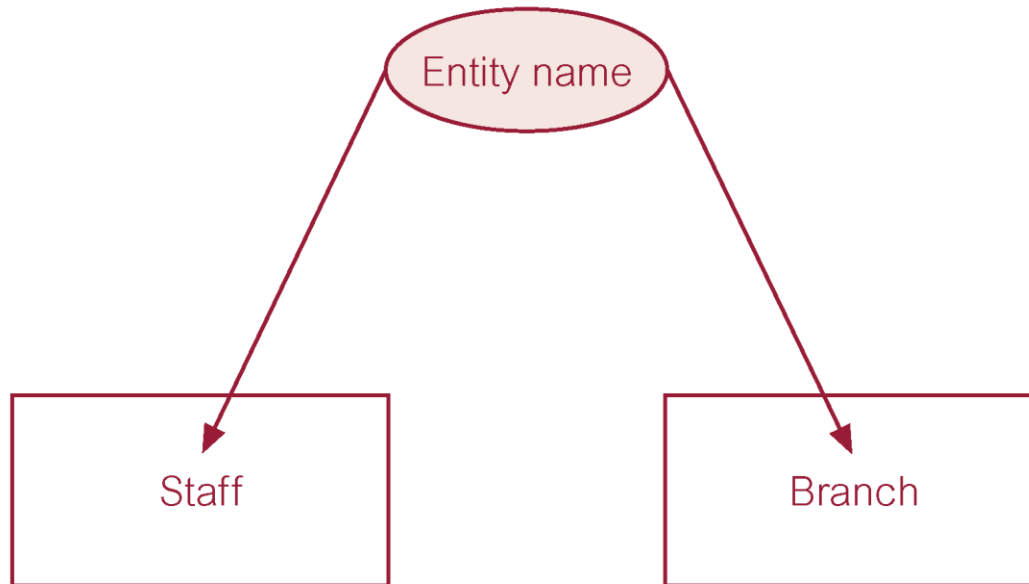
Entity Type

- ▶ **Entity type**
 - **Group of objects with same properties, identified by enterprise as having an independent existence.**
- ▶ **Entity occurrence**
 - **Uniquely identifiable object of an entity type.**

Examples of Entity Types

Physical existence	
Staff	Part
Property	Supplier
Customer	Product
Conceptual existence	
Viewing	Sale
Inspection	Work experience

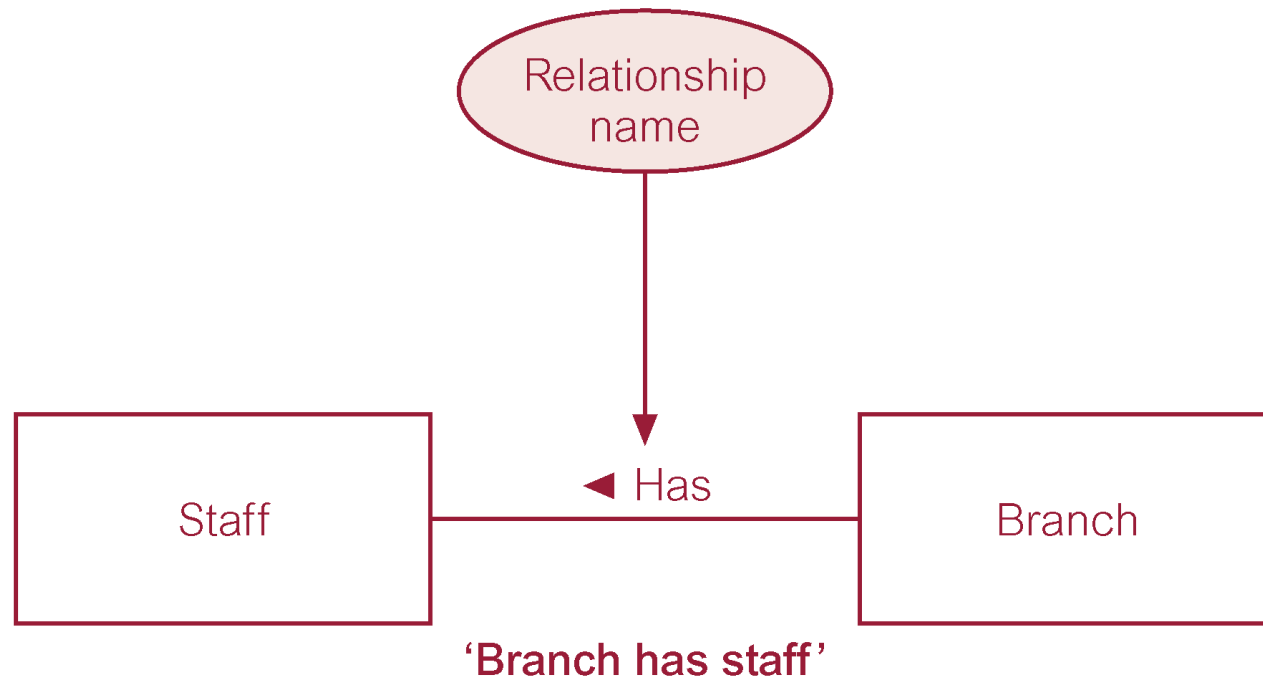
ER Diagram of Staff and Branch Entity Types



Relationship Types

- ▶ **Relationship type**
 - Set of meaningful associations among entity types.
- ▶ **Relationship occurrence**
 - Uniquely identifiable association, which includes one occurrence from each participating entity type.

ER Diagram of Branch *Has* Staff Relationship



Relationship Types

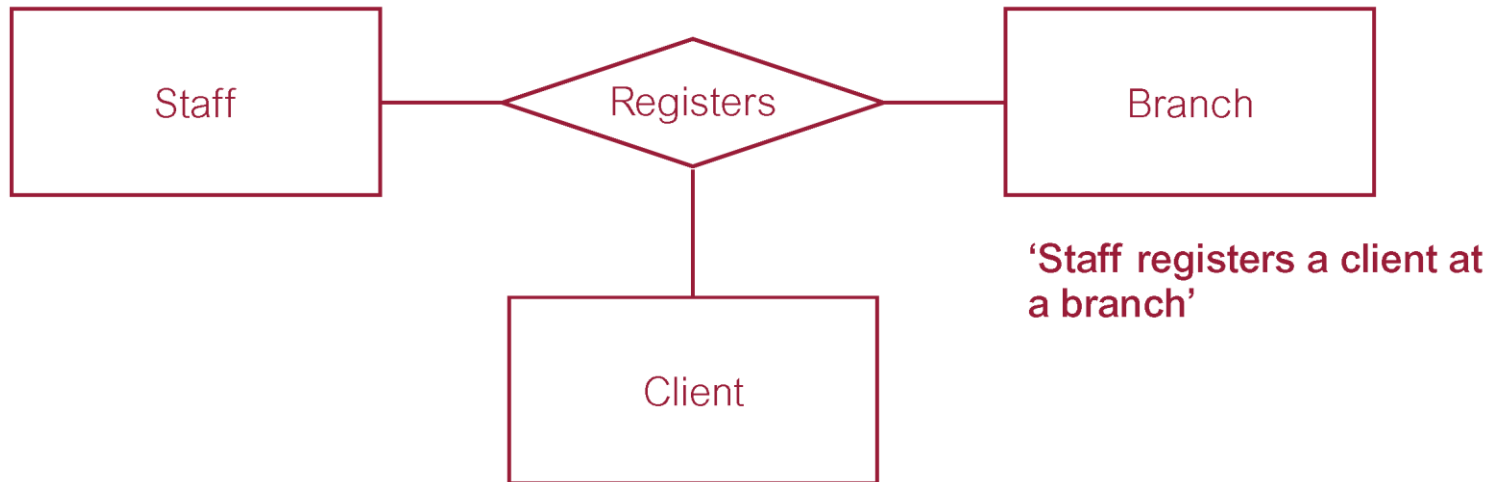
- ▶ **Degree of a Relationship**
 - Number of participating entities in relationship.
- ▶ **Relationship of degree:**
 - two is binary;
 - three is ternary;
 - four is quaternary.

Binary Relationship called *POwns*

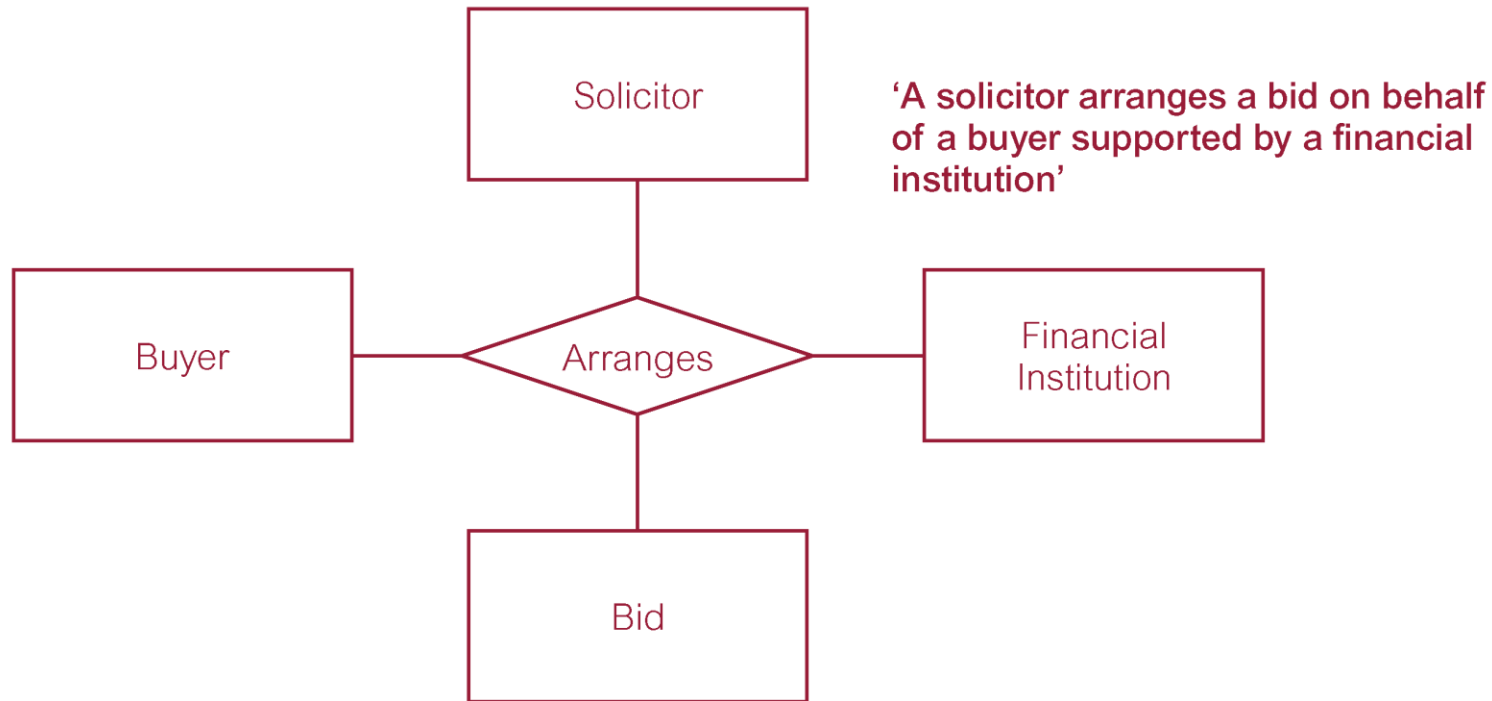
'Private owner owns property for rent'



Ternary Relationship called *Registers*



Quaternary Relationship called *Arranges*



Relationship Types

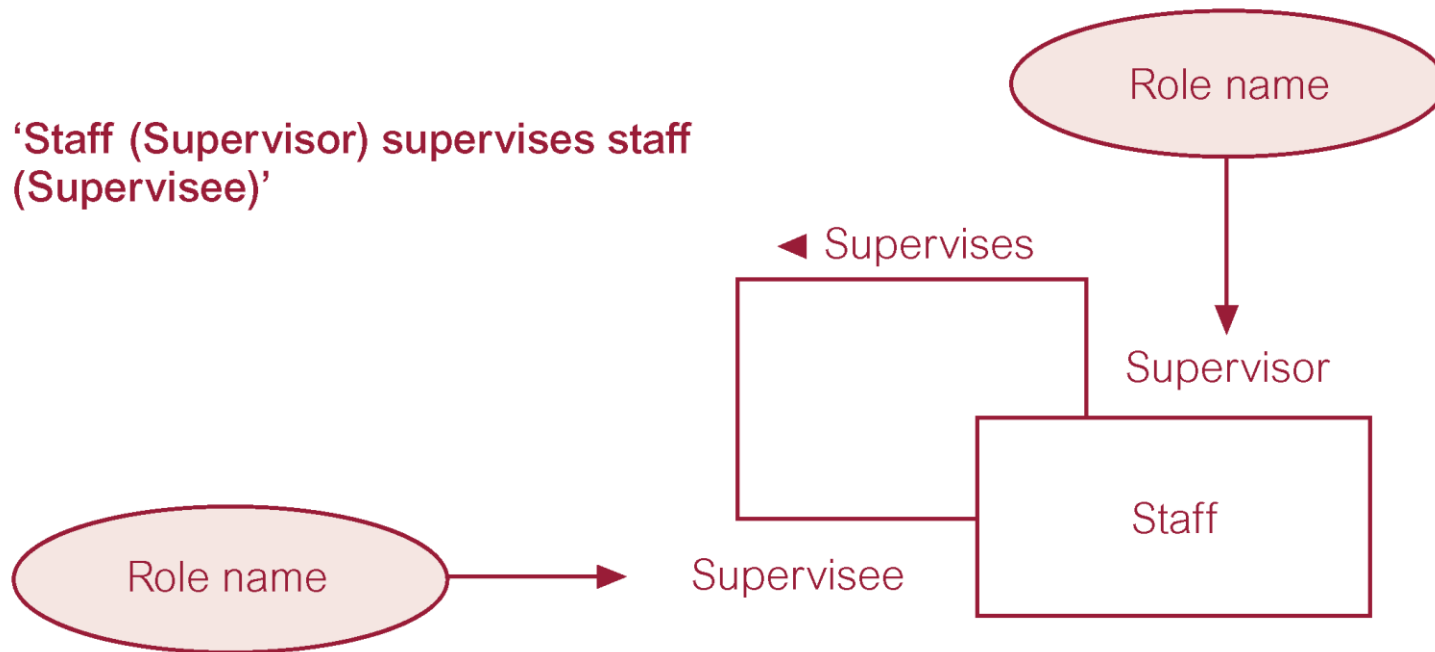
▶ Recursive Relationship

- Relationship type where *same* entity type participates more than once in *different roles*.

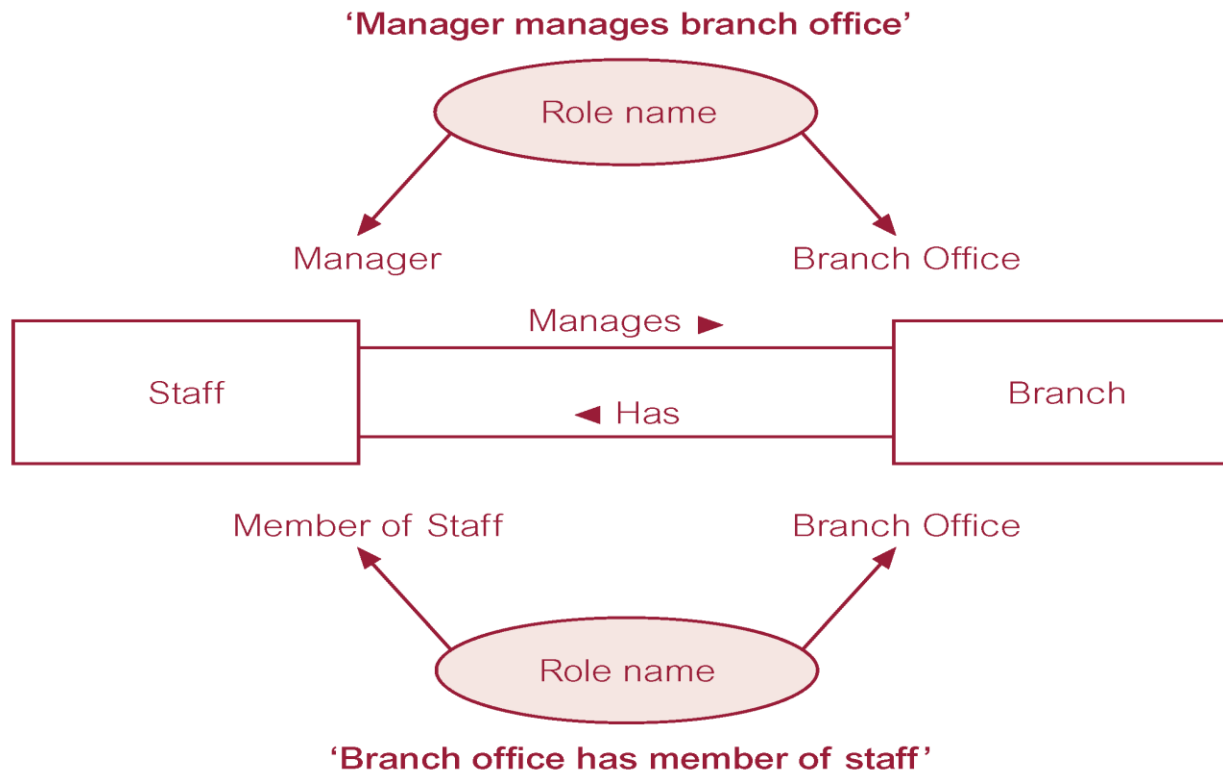
▶ Relationships may be given role names to indicate purpose that each participating entity type plays in a relationship.

Recursive Relationship called *Supervises* with Role Names

'Staff (Supervisor) supervises staff
(Supervisee)'



Entities associated through two distinct Relationships with Role Names



Attributes

- ▶ **Attribute**
 - **Property of an entity or a relationship type.**
- ▶ **Attribute Domain**
 - **Set of allowable values for one or more attributes.**

Attributes

- ▶ **Simple Attribute**

- Attribute composed of a single component with an independent existence.

- ▶ **Composite Attribute**

- Attribute composed of multiple components, each with an independent existence.

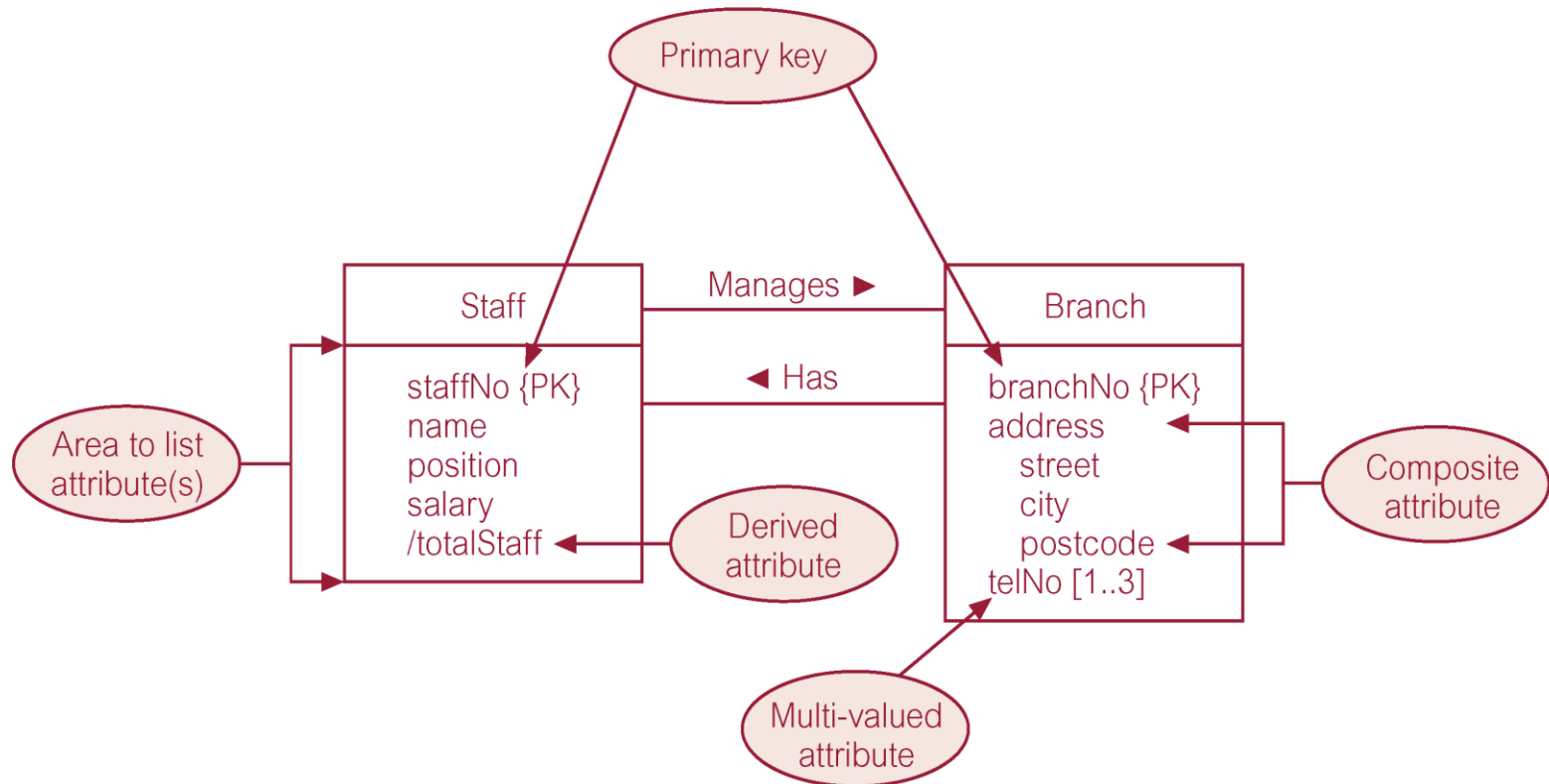
Attributes

- ▶ **Single-valued Attribute**
 - Attribute that holds a single value for each occurrence of an entity type.
- ▶ **Multi-valued Attribute**
 - Attribute that holds multiple values for each occurrence of an entity type.

Keys

- ▶ **Candidate Key**
 - Minimal set of attributes that uniquely identifies each occurrence of an entity type.
- ▶ **Primary Key**
 - Candidate key selected to uniquely identify each occurrence of an entity type.
- ▶ **Composite Key**
 - A candidate key that consists of two or more attributes.

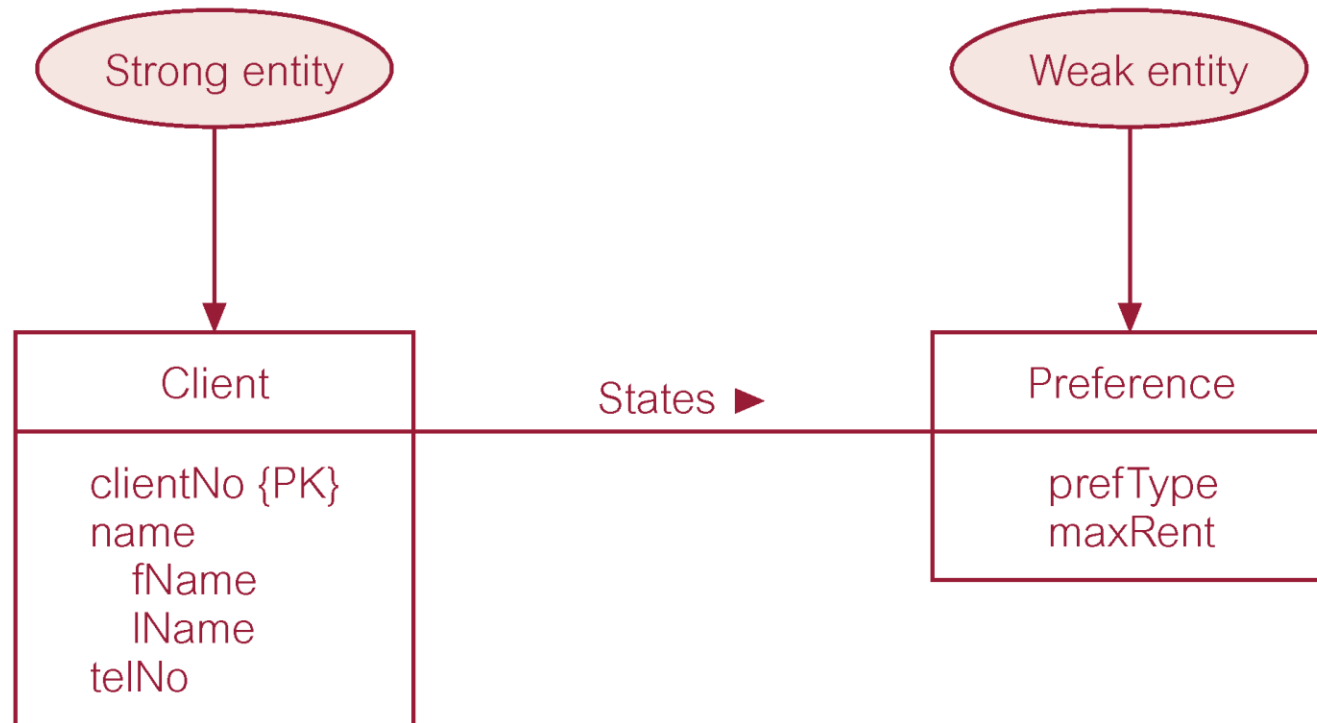
ER Diagram of Staff and Branch Entities and their Attributes



Entity Type

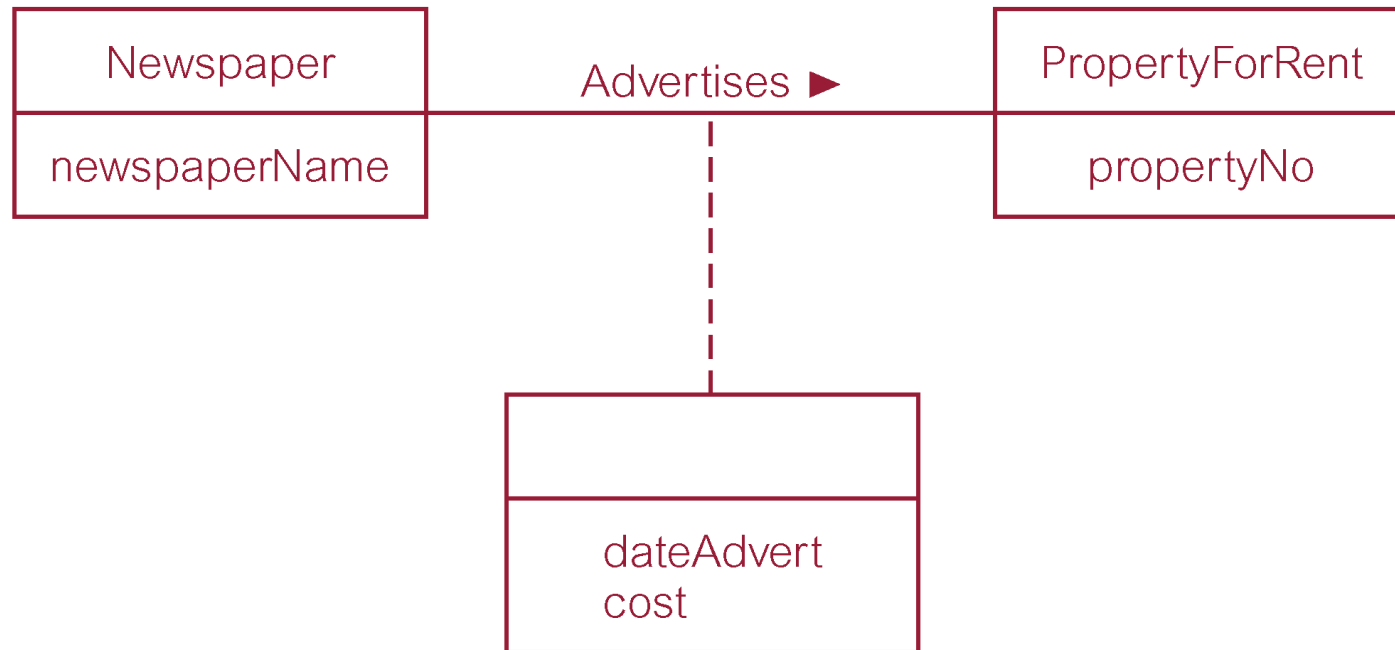
- ▶ **Strong Entity Type**
 - Entity type that is *not* existence-dependent on some other entity type.
- ▶ **Weak Entity Type**
 - Entity type that is existence-dependent on some other entity type.

Strong Entity Type called Client and Weak Entity Type called Preference



Relationship called *Advertises* with Attributes

‘Newspaper advertises property for rent’



Structural Constraints

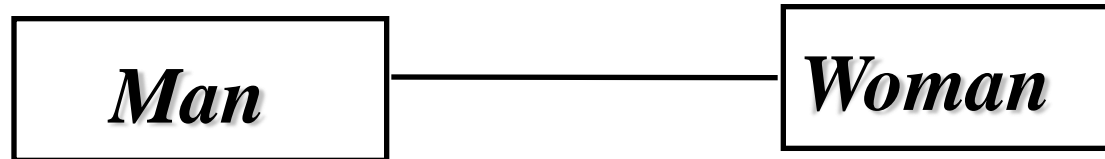
- ▶ Main type of constraint on relationships is called multiplicity.
- ▶ Multiplicity - number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.
- ▶ Represents policies (called *business rules*) established by user or company.

Structural Constraints

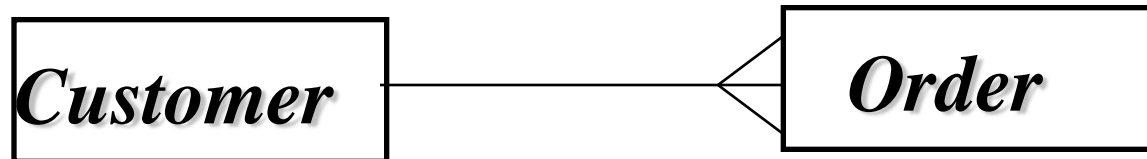
- ▶ **The most common degree for relationships is binary.**
- ▶ **Binary relationships are generally referred to as being:**
 - **one-to-one (1:1)**
 - **one-to-many (1:*) or (1:n)**
 - **many-to-many (*:*) or (m:n)**

Degrees of relationship, alternative representation

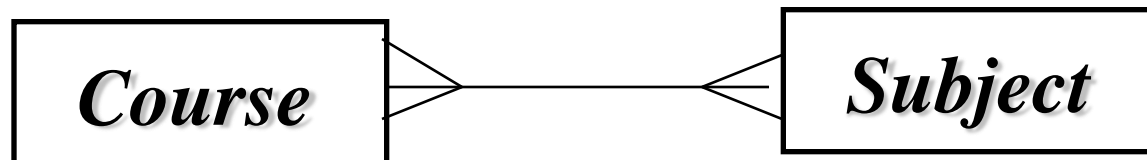
One-to-one (1:1)



One-to-many (1:n)

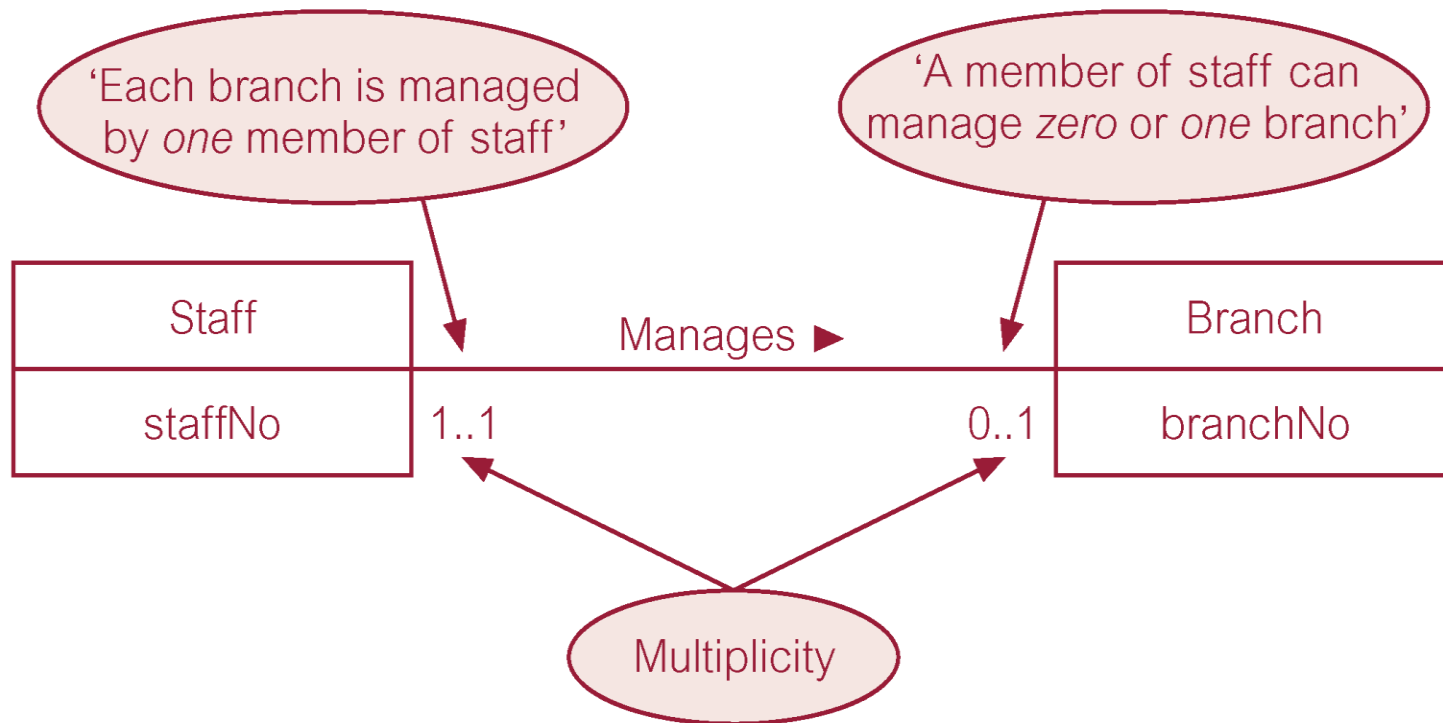


Many-to-many (n:m)



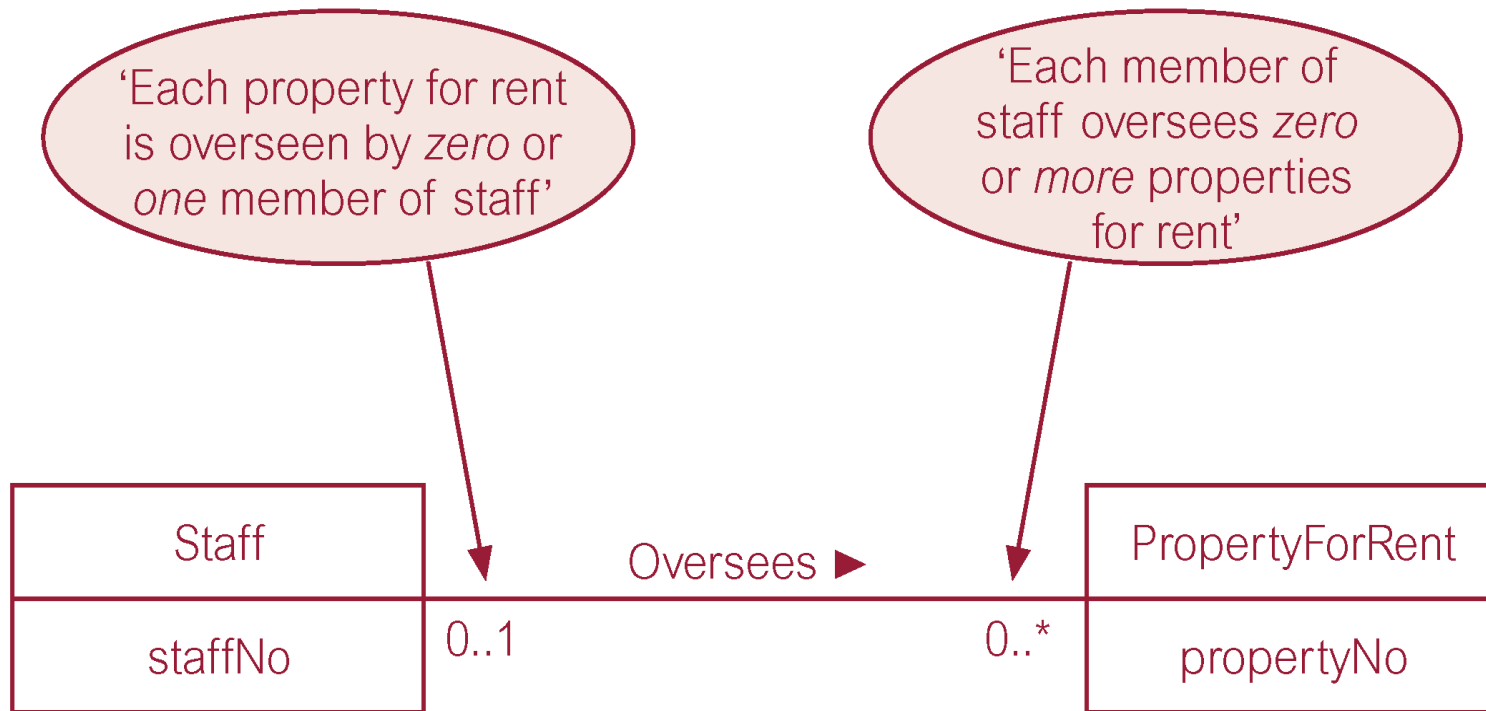
NOTE: Every many to many relationship consists of two one to many relationships working in opposite directions

Multiplicity of Staff *Manages* Branch (1:1) Relationship Type

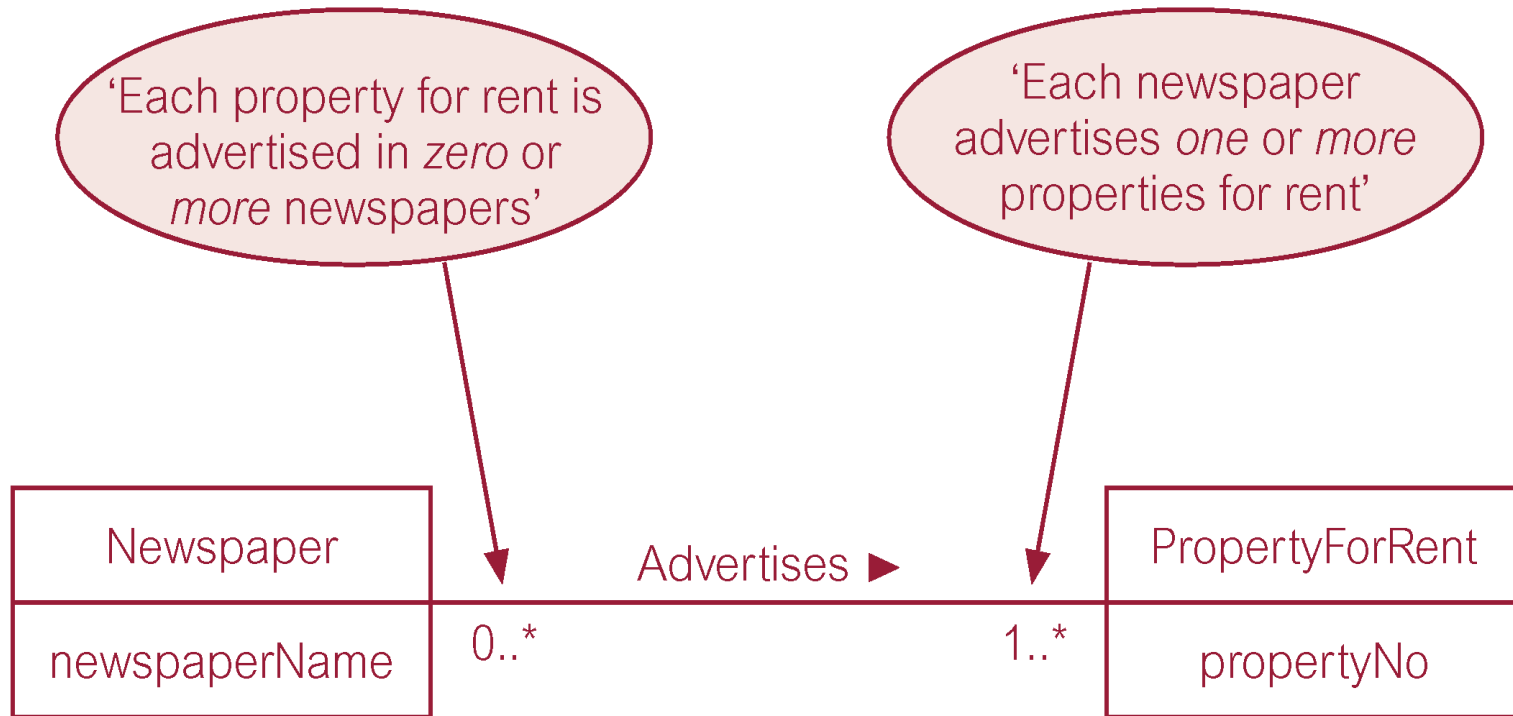


Multiplicity of Staff Oversees PropertyForRent (1:*)

Relationship Type



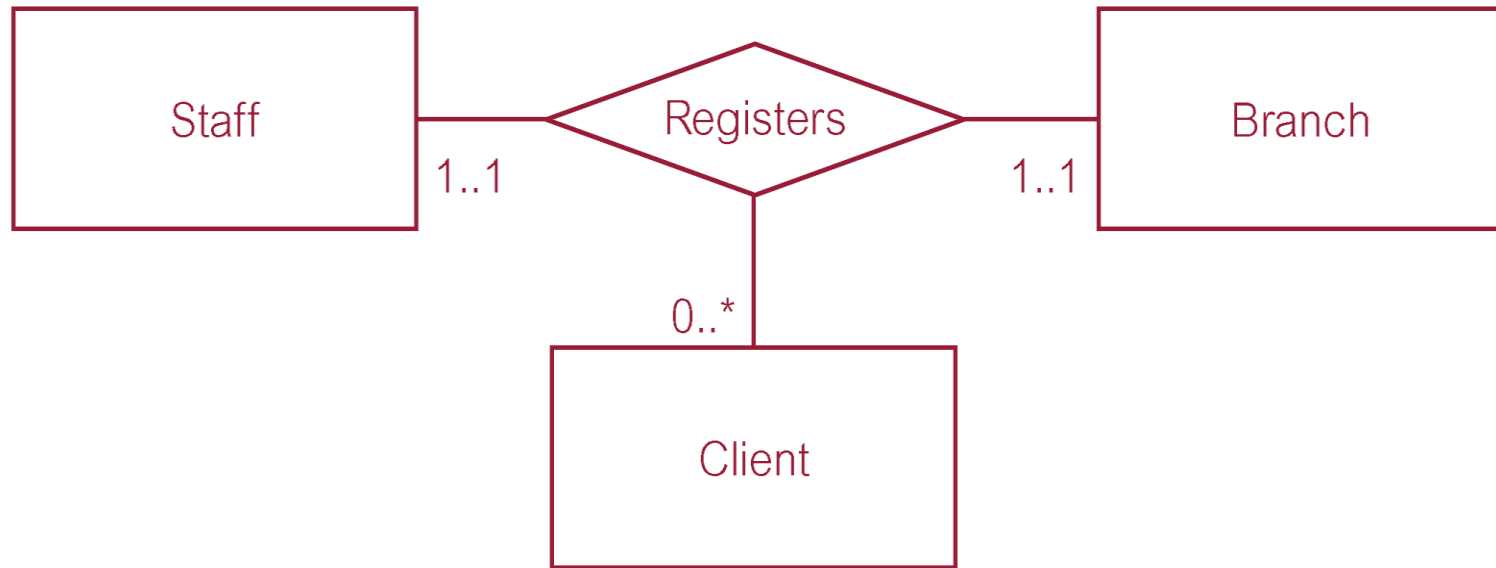
Multiplicity of Newspaper *Advertises* PropertyForRent (*:*) Relationship



Structural Constraints

- ▶ **Multiplicity for Complex Relationships**
 - **Number (or range) of possible occurrences of an entity type in an n -ary relationship when other $(n-1)$ values are fixed.**

Multiplicity of Ternary *Registers* Relationship



Summary of Multiplicity Constraints

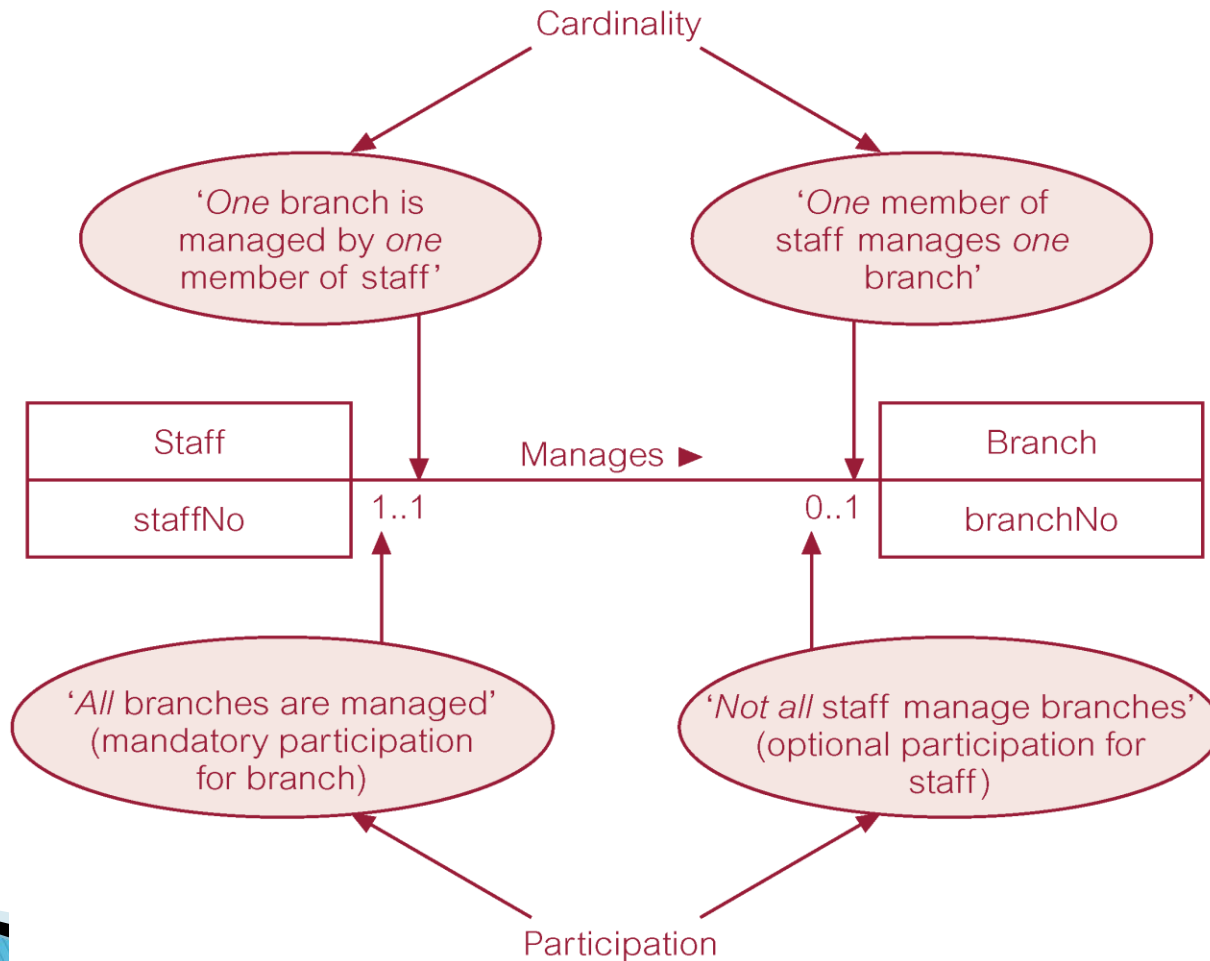
Table 11.1 A summary of ways to represent multiplicity constraints.

Alternative ways to represent multiplicity constraints	Meaning
0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6–8	Zero or three or six, seven, or eight entity occurrences

Structural Constraints

- ▶ **Multiplicity is made up of two types of restrictions on relationships: cardinality and participation.**
- ▶ **Cardinality**
 - **Describes maximum number of possible relationship occurrences for an entity participating in a given relationship type.**
- ▶ **Participation**
 - **Determines whether all or only some entity occurrences participate in a relationship.**

Multiplicity as Cardinality and Participation Constraints



Library Case Study

- ▶ When a library first receives a book from a publisher it is sent, together with the accompanying delivery note, to the library desk. Here the delivery note is checked against a file of books ordered.
- ▶ If no order can be found to match the note, a letter of enquiry is sent to the publishers. If a matching order is found, a catalogue note is prepared from the details on the validated delivery note.
- ▶ The catalogue note, together with the book, is sent to the registration department. The validated delivery note is sent to the accounts department where it is stored.
- ▶ On receipt of an invoice from the publisher, the accounts department checks its store of delivery notes. If the corresponding delivery note is found then an instruction to pay the publishers is made, and subsequently a cheque is sent. If no corresponding delivery note is found, the invoice is stored in a pending file.